

DOCKET NO.: I20 06799 US

PATENT

APPARATUS AND METHOD FOR MODELING
RELATIONSHIPS BETWEEN SIGNALS

Inventors:

Joseph Z. Lu
5631 West Arrowhead Lakes Drive
Glendale, Maricopa County
Arizona 80308
Citizen of the United States

Assignee:

Honeywell International Inc.
101 Columbia Road
Morristown, New Jersey 07962
United States of America

William A. Munck
Davis Munck, P.C.
900 Three Galleria Tower
13155 Noel Road
Dallas, Texas 75240
972-628-3600

APPARATUS AND METHOD FOR MODELING
RELATIONSHIPS BETWEEN SIGNALS

CROSS REFERENCE TO RELATED APPLICATIONS

[001] This patent application is related to U.S. Patent Application No. __/__,__ entitled "APPARATUS AND METHOD FOR ISOLATING NOISE EFFECTS IN A SIGNAL" [Attorney Docket No. I20 06798 US] filed on _____, which is incorporated by reference.

TECHNICAL FIELD

[002] This disclosure relates generally to model identification systems and more specifically to an apparatus and method for modeling relationships between signals.

BACKGROUND

[003] Process control systems are often used to control the operation of a system. For example, a process control system may be used to control the operation of a processing facility. As a particular example, a process control system could manage the use of valves in a processing facility, where the valves control the flow of materials in the facility. Example processing facilities include manufacturing plants, chemical plants, crude oil refineries, and ore processing plants.

[004] Conventional process control systems often use models to predict the behavior of a system being monitored. However, it is often difficult to identify the models used by the process control systems. For example, the conventional process control systems often process signals that suffer from noise or other disturbances. The presence of noise in the signals often makes it difficult for a process control system to identify a relationship between two or more signals. As a result, this often makes it more difficult to monitor and control a system.

SUMMARY

[005] This disclosure provides an apparatus and method for modeling relationships between signals.

[006] In one aspect, a method includes receiving a projection associated with a first signal and a second signal. The second signal includes a first portion associated with the first signal and a second portion not associated with the first signal. The projection at least substantially separates the first portion of the second signal from the second portion of the second signal. The method also includes identifying one or more parameters of a model using at least a portion of the projection. The model associates the first signal and the first portion of the second signal.

[007] In another aspect, an apparatus includes at least one input operable to receive a first signal and a second signal. The second signal includes a first portion associated with the first signal and a second portion not associated with the first signal. The apparatus also includes at least one processor operable to generate a projection associated with the first and second signals and to identify one or more parameters of a model associating the first signal and the first portion of the second signal. The projection at least substantially separates the first portion of the second signal from the second portion of the

second signal.

[008] In yet another aspect, a computer program is embodied on a computer readable medium and is operable to be executed by a processor. The computer program includes computer readable program code for generating a projection associated with a first signal and a second signal. The second signal includes a first portion associated with the first signal and a second portion associated with at least one disturbance. The projection at least substantially separates the first portion of the second signal from the second portion of the second signal. The computer program also includes computer readable program code for identifying one or more parameters of a model associating the first signal and the first portion of the second signal using at least a portion of the projection.

[009] Other technical features may be readily apparent to one skilled in the art from the following figures, descriptions, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[010] For a more complete understanding of this disclosure, reference is now made to the following description, taken in conjunction with the accompanying drawings, in which:

[011] FIGURE 1 illustrates an example system for isolating noise effects in a signal according to one embodiment of this disclosure;

[012] FIGURES 2A through 2C illustrate example signals in the system of FIGURE 1 according to one embodiment of this disclosure;

[013] FIGURES 3A through 3C illustrate example matrices used to represent signals in the system of FIGURE 1 according to one embodiment of this disclosure;

[014] FIGURES 4A through 4E illustrate example projections used to represent signals in the system of FIGURE 1 according to one embodiment of this disclosure;

[015] FIGURE 5 illustrates an example method for isolating noise effects in a signal according to one embodiment of this disclosure;

[016] FIGURES 6A through 6C illustrate an example mechanism for applying canonical QR-decompositions to model identification according to one embodiment of this disclosure; and

[017] FIGURE 7 illustrates an example method for modeling

DOCKET NO.: I20 06799 US

PATENT

relationships between signals according to one embodiment of this disclosure.

DETAILED DESCRIPTION

[018] FIGURE 1 illustrates an example system 100 for isolating noise effects in a signal according to one embodiment of this disclosure. The system 100 shown in FIGURE 1 is for illustration only. Other embodiments of the system 100 may be used without departing from the scope of this disclosure.

[019] In this example embodiment, the system 100 includes a monitored system 102. The monitored system 102 represents any suitable system for producing or otherwise receiving an input signal 104 and producing or otherwise providing an ideal output signal 106. In some embodiments, the monitored system 102 is represented by a process model $G(s)$, which represents the transformation of the input signal 104 into the output signal 106.

[020] The monitored system 102 may represent any type of system. The monitored system 102 could, for example, represent a manufacturing or other processing system or a communication system. As a particular example, the monitored system 102 could represent a manufacturing plant having various valves that are controlled based on the input signal 104 and/or the ideal output signal 106. The monitored system 102 could also represent a communication system where the input signal 104 represents a signal transmitted by a mobile telephone and the ideal output

signal 106 represents the ideal signal to be received by a base station.

[021] As shown in FIGURE 1, the ideal output signal 106 is often corrupted by some type of noise or other disturbance 108. This leads to the creation of an actual output signal 110. The actual output signal 110 includes a first portion associated with the input signal 104 and a second portion associated with the noise 108. These two portions often overlap, making it difficult to separate them. The noise or other disturbance 108 could represent any suitable disturbance to the ideal output signal 106, such as white noise or colored noise. As a particular example, the monitored system 102 could represent a production system, and the noise 108 could represent white noise introduced into an ideal output signal 106 before the signal 106 is received by a valve controller.

[022] In the example in FIGURE 1, the system 100 includes a controller 112, which has access to the input signal 104 and the actual output signal 110. The controller 112 uses the input signal 104 and the actual output signal 110 to control the operation of the monitored system 102. For example, the controller 112 could represent a valve controller capable of controlling the opening and closing of valves in the monitored system 102. As another example, the controller 112 could

represent a signal controller capable of analyzing the input signal 104 and the actual output signal 110 and adjusting one or more parameters used to transmit data in the system 100. The controller 112 includes any hardware, software, firmware, or combination thereof for controlling one or more aspects of operation of the system 100. As a particular example, the controller 112 could include one or more processors 114 and one or more memories 116 capable of storing data and instructions used by the processors. In this example, the controller 112 receives the input signal 104 through a first input 118 and the actual output signal 110 through a second input 120.

[023] As shown in FIGURE 1, the controller 112 only has access to an output signal 110 that has been altered because of noise or other disturbances 108. Conventional systems attempt to remove noise or other disturbances 108 from a signal 110 using low-pass filtering. Low-pass filters often cannot eliminate much of the noise or other disturbances 108 from a signal 110 without impeding the performance of the system 100.

[024] To facilitate more accurate control over the monitored system 102, the controller 112 generates at least one matrix associated with the input signal 104 and the actual output signal 110. The controller 112 then generates a projection of the matrix using "canonical QR-decomposition." This projects the

matrix into orthogonal space, where the projection at least partially separates the input signal 104, the portion of the actual output signal 110 corresponding to the input signal 104, and the portion of the actual output signal 110 corresponding to the noise or other disturbances 108. In this way, the controller 112 at least partially separates the effects of the input signal 104 in the output signal 110 from the effects of the noise 108 in the output signal 110. As a result, the controller 112 is able to more effectively isolate the effects of noise 108 in the actual output signal 110.

[025] QR-decomposition refers to a matrix decomposition performed according to the following equation:

$$A = QR$$

where A represents a matrix being decomposed, Q represents an orthogonal matrix, and R represents an upper triangular matrix.

[026] A problem with conventional QR-decomposition is that a given matrix A could be decomposed in different ways. For example, a given matrix A could be decomposed into $[Q_1 R_1]$, $[Q_2 R_2]$, or $[Q_3 R_3]$. This creates problems in isolating noise 108 in the actual output signal 110 because it means that the same matrix representing the same input signals 104 and actual output signals 110 could have different QR-decompositions.

[027] Canonical QR-decomposition or "CQR decomposition"

represents a unique QR-decomposition where the diagonal values in the triangular matrix R are greater than or equal to zero. The "diagonal values" in the matrix R represent the values along the diagonal between the upper left corner and the lower right corner of the matrix R . By preventing the diagonal values in the upper triangular matrix R from being less than zero, each matrix A can be uniquely decomposed. This helps to facilitate the separation of noise effects contained in the actual output signal 110. In some embodiments, software routines are used to decompose a matrix using canonical QR-decomposition. Example software to decompose a matrix using canonical QR-decomposition is shown in the Software Appendix.

[028] Although FIGURE 1 illustrates one example of a system 100 for isolating noise effects in a signal, various changes may be made to FIGURE 1. For example, the functionality of the controller 112 could be implemented in any hardware, software, firmware, or combination thereof. Also, the functionality of the controller 112 could be used in any other apparatus, system, or environment. As particular examples, the functionality of the controller 112 could also be implemented in a monitor, modeling tool, evaluator, detector, adapter, or any other device or system.

[029] FIGURES 2A through 2C illustrate example signals in the

system 100 of FIGURE 1 according to one embodiment of this disclosure. The signals shown in FIGURES 2A through 2C are for illustration only. The system 100 of FIGURE 1 or other system may receive, produce, or otherwise have access to any other signals without departing from the scope of this disclosure.

[030] FIGURE 2A illustrates an example input signal 104 received by the monitored system 102 in FIGURE 1. In particular, FIGURE 2A plots the values of five hundred samples of the input signal 104. As shown in FIGURE 2A, the input signal 104 may vary widely over a small number of samples and over a longer period of time.

[031] FIGURE 2B illustrates an example ideal output signal 106 produced or otherwise provided by the monitored system 102 in FIGURE 1. In particular, FIGURE 2B plots the values of five hundred samples of the ideal output signal 106. As shown in FIGURE 2B, this particular ideal output signal 106 varies but not as rapidly or widely as the input signal 104. Also, the ideal output signal 106 does not appear to include random peaks or valleys, which often indicate the presence of noise.

[032] FIGURE 2C illustrates an example actual output signal 110 produced or otherwise provided by the monitored system 102 in FIGURE 1. In particular, FIGURE 2C plots the values of five hundred samples of the actual output signal 110. As shown in

FIGURE 2C, the actual output signal 110 includes random peaks and valleys, indicating that the actual output signal 110 has been corrupted by noise or other disturbances 108.

[033] The controller 112 or other monitor in the system 100 of FIGURE 1 often has access only to the input signal 104 and the actual output signal 110. The controller 112 or other monitor generally lacks access to the ideal output signal 106. As shown in FIGURE 2C, it is typically difficult to estimate or extract the ideal output signal 106 from the actual output signal 110. For example, running the actual output signal 110 through a low-pass filter could remove much, but not all, of the noise and also remove some of the ideal output signal 106.

[034] As described above, the controller 112 separates the effects of noise 108 from the effects of the input signal 104 in the output signal 110. In particular, the controller 112 generates a matrix and performs canonical QR-decomposition to project the matrix into orthogonal space, where the input signal 104, the portion of the actual output signal 110 corresponding to the input signal 104, and the portion of the actual output signal 110 corresponding to the noise 108 are at least partially separated. In this way, the controller 112 or other monitor can at least partially separate the noise effects from the input effects in the actual output signal 110.

[035] Although FIGURES 2A through 2C illustrate different examples of the signals in the system 100 of FIGURE 1, various changes may be made to FIGURES 2A through 2C. For example, any other or additional signals may be present in the system 100 of FIGURE 1 or in any other suitable system. Also, the various signals may represent any suitable information, such as information related to the operation of a valve in a production facility.

[036] FIGURES 3A through 3C illustrate example matrices used to represent signals in the system of FIGURE 1 according to one embodiment of this disclosure. The matrices shown in FIGURES 3A through 3C are for illustration only. The system 100 of FIGURE 1 or other system may produce or otherwise use any other matrix or matrices without departing from the scope of this disclosure.

[037] A matrix 300 in FIGURE 3A contains samples 302 from the actual output signal 110. As shown in FIGURE 3A, each row of the matrix 300 includes k samples 302 of the actual output signal 110, and each column of the matrix 300 includes $n-(k+1)$ samples 302 of the actual output signal 110. In particular embodiments, the number of rows in the matrix 300 is much greater than the number of columns in the matrix 300, although any suitable number of rows and/or columns may be used.

[038] At least some of the samples 302 of the actual output

signal 110 appear multiple times in the matrix 300. For example, the sample 302 labeled " y_2 " appears twice in a diagonal pattern, and the sample 302 labeled " y_3 " appears three times in a diagonal pattern. Overall, the matrix 300 includes n different samples 302 of the actual output signal 110.

[039] In this example, the matrix 300 represents a "column Hankel matrix." In this type of matrix, the matrix includes a time series of samples 302 in the horizontal direction 304 (left to right) and a time series of samples 302 in the vertical direction 306 (top to bottom). Because the samples 302 in the horizontal direction 304 form a time series in the left-to-right direction, the matrix 300 represents a "forward" column Hankel matrix.

[040] A different matrix 330 is shown in FIGURE 3B, which contains samples 332 of the input signal 104. Each row includes k samples, and each column includes $n-(k+1)$ samples. As with the matrix 300 in FIGURE 3A, the matrix 330 in FIGURE 3B represents a column Hankel matrix. The matrix 330 includes a time series of samples 332 in the horizontal direction 334 and a time series of samples 332 in the vertical direction 336. However, the samples 332 in the matrix 330 represent a time series of samples 332 in the opposite horizontal direction 334 (right to left), so the matrix 330 represents a "backward" column Hankel matrix.

[041] To isolate the effects of noise 108 in the actual output signal 110 from the effects of the input signal 104, the controller 112 may generate the matrices 300, 330 using the samples 302, 332 of the actual output signal 110 and the input signal 104. The controller 112 then generates a matrix 360, which is shown in FIGURE 3C. The matrix 360 includes both the backward column Hankel matrix 330 representing the input signal 104 and a forward column Hankel matrix 300 representing the actual output signal 110. After generating the matrix 360, the controller 112 or other monitor decomposes the matrix 360 using CQR decomposition to project the matrix 360 into orthogonal space. The projection at least partially separates the noise effects from the input effects in the actual output signal 110.

[042] Although FIGURES 3A through 3C illustrate example matrices used to represent signals in the system of FIGURE 1, various changes may be made to FIGURES 3A through 3C. For example, FIGURE 3A shows a forward column Hankel matrix and FIGURE 3B shows a backward column Hankel matrix. The matrix 300 in FIGURE 3A could be formatted as a backward column Hankel matrix and/or the matrix 330 in FIGURE 3B could be formatted as a forward column Hankel matrix.

[043] FIGURES 4A through 4E illustrate example projections used to represent signals in the system of FIGURE 1 according to

one embodiment of this disclosure. The projections shown in FIGURES 4A through 4E are for illustration only. The system 100 of FIGURE 1 or other system may produce or otherwise use any other projections without departing from the scope of this disclosure.

[044] FIGURE 4A illustrates a projection 400 associated with a matrix 360, where the left portion of the matrix 360 represents a backward column Hankel matrix of the input signal 104 and the right portion represents a forward column Hankel matrix of the ideal output signal 106. In this example, the matrix 360 is denoted using the notation:

$$[U_b \hat{Y}]$$

where U represents a column Hankel matrix of the input signal 104, \hat{Y} represents a column Hankel matrix of the ideal output signal 106, and b indicates that a matrix is a backward column Hankel matrix. By default, any matrix without a b sub-notation represents a forward column Hankel matrix.

[045] In this example, the matrix 360 is decomposed using CQR decomposition so as to project the matrix 360 into orthogonal space. The orthogonal space is defined by three axes 402, 404, 406. The first axis 402 represents an index of the rows in the decomposed matrix, and the second axis 404 represents an index of the columns in the decomposed matrix. Both indexes increase

moving from left to right in FIGURE 4A. The third axis 406 represents the values contained in the decomposed matrix 360.

[046] As shown in FIGURE 4A, the projection 400 of the matrix 360 includes two different portions 408 and 410. The first portion 408 represents the input signal 104, and the second portion 410 represents the ideal output signal 106. Because the second portion 410 represents the ideal output signal 106, the second portion 410 represents only the effects of the input signal 104 without any effects of noise or other disturbances 108.

[047] In contrast, FIGURE 4B illustrates a projection 420 associated with a matrix 360, where the left portion of the matrix 360 represents a backward column Hankel matrix of the input signal 104 and the right portion represents a forward column Hankel matrix of the actual output signal 110. In this example, the matrix 360 is denoted using the notation:

$$[U_b \ Y]$$

where Y represents a column Hankel matrix of the actual output signal 110.

[048] In this example, the matrix 360 is decomposed using CQR decomposition so as to project the matrix 360 into the same orthogonal space. As shown in FIGURE 4B, the projection 420 of the matrix 360 includes three different portions 428, 430, 432.

The first portion 428 represents the input signal 104. The second portion 430 substantially represents the portion of the actual output signal 110 caused by the input signal 104. In other words, the second portion 430 of the projection 420 substantially represents the ideal output signal 106. The third portion 432 substantially represents the noise 108 contained in the actual output signal 110. Because the projection 420 substantially separates the response of the system 102 to the input signal 104 from the effects of noise 108, the controller 112 may more accurately process the actual output signal 110.

[049] FIGURES 4A and 4B illustrate projections of matrices 360 that include a backward column Hankel matrix on the left side and a forward column Hankel matrix on the right side. Other matrices could be produced and then decomposed according to particular needs. For example, FIGURE 4C illustrates a projection 440 associated with a matrix 360, where the left portion of the matrix 360 represents a forward column Hankel matrix of the input signal 104 and the right portion represents a backward column Hankel matrix of the actual output signal 110. In this example, the matrix 360 is denoted using the notation:

$$[U \ Y_b].$$

[050] In this example, the matrix 360 is decomposed using CQR decomposition so as to project the matrix 360 into the orthogonal

space. As shown in FIGURE 4C, the projection 440 of the matrix 360 includes three different portions 448, 450, 452. The first portion 448 represents the input signal 104. The second and third portions 450, 452 represent the portion of the actual output signal 110 caused by the input signal 104 and the portion of the actual input signal 110 caused by noise 108. However, the second and third portions 450, 452 are interlaced.

[051] Similarly, FIGURE 4D illustrates a projection 460 associated with a matrix 360, where the left portion of the matrix 360 represents a forward column Hankel matrix of the input signal 104 and the right portion represents a forward column Hankel matrix of the actual output signal 110. In this example, the matrix 360 is denoted using the notation:

$$[U \ Y].$$

[052] In this example, the matrix 360 is decomposed using CQR decomposition so as to project the matrix 360 into the orthogonal space. As shown in FIGURE 4D, the projection 460 of the matrix 360 includes four different portions 468, 470a-470b, 472. The first portion 468 represents the input signal 104. The second and third portions 470a-470b substantially represent the portion of the actual output signal 110 caused by the input signal 104. As shown in FIGURE 4D, the portion of the actual output signal 110 caused by the input signal 104 has been dissected into two

different parts 470a and 470b. The fourth portion 472 substantially represents the noise 108 contained in the actual output signal 110.

[053] Finally, FIGURE 4E illustrates a projection 480 associated with a matrix 360, where the left portion of the matrix 360 represents a backward column Hankel matrix of the input signal 104 and the right portion represents a backward column Hankel matrix of the actual output signal 110. In this example, the matrix 360 is denoted using the notation:

$$[U_b \ Y_b].$$

[054] In this example, the matrix 360 is decomposed using CQR decomposition so as to project the matrix 360 into the orthogonal space. As shown in FIGURE 4E, the projection 480 of the matrix 360 includes three different portions 488, 490, 492. The first portion 488 represents the input signal 104. The second and third portions 490, 492 represent the portion of the actual output signal 110 caused by the input signal 104 and the portion of the actual input signal 110 caused by noise 108. However, the second and third portions 490, 492 are interlaced.

[055] Using one or more of these projections, the controller 112 or other monitor in the system 100 of FIGURE 1 may more effectively process the various signals. For example, the controller 112 or other monitor could use the projection 420 in

FIGURE 4B to identify the effects of the input signal 104 in the actual output signal 110 and the effects of the noise 108 in the actual output signal 110. The controller 112 or other monitor could use this information in any suitable manner. For example, the controller 112 could disregard the effects of the noise 108 in the actual output signal 110 and process only the effects of the input signal 104 in the actual output signal 110. As another example, the controller 112 or other monitor could use this information to identify relationships between the input and output signals.

[056] As can be seen in FIGURE 3C, the matrix 360 used to form the projections shown in FIGURES 4A through 4E could become very large as the number of samples grows. For example, if each signal is represented by 10,000 samples, each matrix 300, 330 might have 1,000 columns (k) and 8,999 rows ($n-(k+1)$), and the matrix 360 would have 2,000 columns and 8,999 rows.

[057] In some embodiments, to reduce the processing power and time needed by the controller 112 to process the signals, the controller 112 processes the samples in batches. For example, the controller 112 could process samples of the input signal 104 and actual output signal 110 in batches of five hundred samples each.

[058] To help reduce the size of the matrix needed to

generate a projection, the controller 112 may generate and process a first matrix 360 associated with a first batch of the samples. The first matrix 360 is decomposed into Q_1 and R_1 . To process the next batch of samples, the controller 112 generates a matrix 360 for the next batch of samples and combines that matrix 360 with R_1 . For example, the controller 112 could combine a new matrix 360 with a previous R matrix to create a concatenated matrix as follows:

$$\begin{bmatrix} R_{x-1} \\ Data_x \end{bmatrix}$$

where x represents the number of the current data segment (where $x \geq 2$), $Data_x$ represents the data samples in the x -th data segment, and R_{x-1} represents the R matrix associated with the $(x-1)$ -th data segment. The matrix resulting from this combination is then processed by the controller 112 and decomposed. This allows the controller 112 to process a smaller matrix, even as the total number of samples becomes very large.

[059] In the example above, the samples in the previous data segments are continuously carried through the processing of future data segments. In effect, the controller 112 is concatenating the data segments together, and the projection corresponding to the x -th data segment represents all previous data segments. In other embodiments, the samples in previous

data segments may be phased out of the processing of future data segments. In effect, this provides a "forgetting factor" where older data segments contribute less to the projection than newer data segments. For example, the controller 112 could combine a new matrix 360 with a previous R matrix as follows:

$$\begin{bmatrix} \lambda \times R_{x-1} \\ Data_x \end{bmatrix}$$

where λ represents a value between zero and one. A λ value of one would operate as described above. A λ value of zero causes the controller 112 to ignore the previous R matrix and only process the current data segment. A λ value between zero and one causes the controller 112 to partially consider the previous R matrix in forming the projection, which over time reduces the effects of older data segments to a greater and greater extent.

[060] Although FIGURES 4A through 4E illustrate examples of different projections used to represent signals in the system 100 of FIGURE 1, various changes may be made to FIGURES 4A through 4E. For example, the projections shown in FIGURES 4A through 4E are for illustration only. Other signals in the system 100 or other system would have different projections.

[061] FIGURE 5 illustrates an example method 500 for isolating noise effects in a signal according to one embodiment of this disclosure. For ease of explanation, the method 500 is

described with respect to the controller 112 operating in the system 100 of FIGURE 1. The method 500 could be used by any other apparatus or device in any system.

[062] The controller 112 receives samples of an input signal at step 502. This may include, for example, the controller 112 receiving samples of an input signal 104 or the controller 112 receiving the input signal 104 and generating the samples.

[063] The controller 112 receives samples of an actual output signal at step 504. This may include, for example, the controller 112 receiving samples of an actual output signal 110 or the controller 112 receiving the actual output signal 110 and generating the samples.

[064] The controller 112 generates a first matrix using the samples of the input signal at step 506. This may include, for example, the controller 112 generating a forward or backward column Hankel matrix 330 using the samples of the input signal 104.

[065] The controller 112 generates a second matrix using the samples of the actual output signal at step 508. This may include, for example, the controller 112 generating a forward or backward column Hankel matrix 300 using the samples of the actual output signal 110.

[066] The controller 112 generates a third matrix using the

first and second matrices at step 510. This may include, for example, the controller 112 generating a third matrix 360 by concatenating the first and second matrices 300, 330.

[067] The controller 112 projects the third matrix into orthogonal space at step 512. This may include, for example, the controller 112 performing CRQ decomposition to project the third matrix 360 into orthogonal space. This may also include the controller 112 generating a projection as shown in FIGURES 4A through 4E.

[068] At this point, the controller 112 may use the projection in any suitable manner. For example, the controller 112 could use the projection to identify a model that relates the input signal 104 to the ideal output signal 106 contained in the actual output signal 110.

[069] Although FIGURE 5 illustrates one example of a method 500 for isolating noise effects in a signal, various changes may be made to FIGURE 5. For example, the controller 112 could generate the third matrix at step 510 directly after the samples are collected at steps 502, 504. In this example, the controller 112 need not generate the first and second matrices at steps 506, 508.

[070] FIGURES 6A through 6C illustrate an example mechanism for applying canonical QR-decompositions to model identification

according to one embodiment of this disclosure. The canonical QR-decompositions of a $[U_b \ Y]$ matrix shown in FIGURES 6A through 6C are for illustration only. Other canonical QR-decompositions may be used without departing from the scope of this disclosure.

[071] In general, the controller 112 may perform model identification to model the behavior of the monitored system 102. The monitored system 102 typically may be represented in many different forms. In a particular embodiment, the monitored system 102 is modeled using a state-space model of the form:

$$x_{k+1} = A \cdot x_k + B \cdot u_k$$

$$y_k = C \cdot x_k + D \cdot u_k$$

where u represents samples of the input signal 104, x represents the states of the monitored system 102, y represents the output of the system 102, and $\{A, B, C, D\}$ are matrices that represent the parameters of the system 102. In this embodiment, the controller 112 performs model identification by determining values for $\{A, B, C, D\}$.

[072] In some embodiments, to perform model identification, the controller 112 generates a projection 420 as shown in FIGURE 4B using the signals 104, 110 associated with the monitored system 102. In particular embodiments, the controller 112 identifies values for A , B , C , and D by selecting one or more regions from the projection. Using the selected region(s), the

controller 112 identifies pole candidates (values for A and C) and model candidates (values for B and D) for the monitored system 102. The controller 112 then performs model validation and order reduction by selecting poles and an order for the monitored system 102. The selected poles and order are used as the model of the monitored system 102.

[073] FIGURE 6A illustrates a canonical QR-decomposition that produces an orthogonal matrix 602 and an upper triangular matrix 604. As shown in FIGURE 6A, the upper triangular matrix 604 includes two diagonals 606a and 606b. As described above, the values along the diagonal 606a are each greater than or equal to zero. The diagonals 606 divide the upper triangular matrix 604 into upper, lower, left, and right sections.

[074] In some embodiments, to identify possible poles of the monitored system 102, the controller 112 defines one or more areas 608a-608c in the upper triangular matrix 604. Although FIGURE 6A illustrates three different areas 608a-608c, any number of areas 608 could be defined. In particular embodiments, using the one or more areas 608a-608c, the controller 112 identifies the possible poles using the following algorithm:

```
[V,S,U] = svd(R2',0)
U1 = U(:,1:n)
[Ng,n] = size(U1)
g = U1 * diag(sqrt(ss(1:n)))
gm = g(1:Ng - Nout, :)
C = g(1:Nout, :)
```

$$A = g_m \backslash g(N_{out} + 1:N_g, :)$$
$$Poles = eig(A).$$

In this algorithm, $[V, S, U]$ represents V , S , and U matrices produced using singular value decomposition (the *svd* function call). U_1 represents the values along the left-most n columns of the U matrix. The value n represents an order of the monitored system 102 and may be specified by the user, determined by thresholding the singular values in the S matrix, or determined in any other suitable manner. N_g represents the number of rows in U_1 . N_{out} represents the number of outputs in the monitored system 102. The variable g represents an observability matrix. The variable g_m represents a shortened observability matrix. A and C are part of the parameter matrices for representing the monitored system 102. The variable $Poles$ represents the eigenvalues of the matrix A , which are the possible poles of the model. In general, if multiple areas 608 are used with the above algorithm, the number of possible poles candidate increases.

[075] Once candidates for the poles (A and C) of the model have been identified, the controller 112 identifies the model candidates (B and D). As shown in FIGURE 6B, for the different pole candidates $[C \ A]$, different matrices U_{fb} 610a-610c can be identified in the upper triangular matrix 604. The bottom right corner of each of these matrices 610a-610c lies on the diagonal 606a and is parallel to the top of the corresponding R_2 area

608a-608c. The size of the matrices 610a-610c can then be selected based on the corresponding size of R_2 .

[076] Each of these matrices 610a-610c can be rewritten from a backward matrix U_{fb} into a forward matrix U_f . The B and D values of the model may then be determined using the following formula:

$$\min_{B,D} \|(I - U_1 U_1') R_2' - L(B,D) U_f\|_2^2$$

where U_1 represents the matrix used above to find the pole candidates, I represents an identity matrix, and $L(B,D)$ represents a matrix defined as a function of B and D.

[077] In particular embodiments, the $L(B,D)$ matrix has the following format:

$$L(B,D) = \begin{bmatrix} D & 0 \\ \Gamma_{N-1} B & H_{N-1} \end{bmatrix} - \begin{bmatrix} C \Gamma_N^\psi H_N \\ \Gamma_{N-1}^\psi A \Gamma_N^\psi H_N \end{bmatrix}$$

where Γ_x represents an order-x extended observability matrix, H_x represents an order-x block impulse response matrix, and ψ denotes a pseudo-inverse. Examples of Γ_x and H_x include:

$$\Gamma_x = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{x-1} \end{bmatrix}$$

and

$$H_x = \begin{bmatrix} D & 0 & 0 & \cdots & 0 \\ CB & D & 0 & \cdots & 0 \\ CAB & CB & D & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ CA^{x-2}B & CA^{x-3}B & CA^{x-4}B & \cdots & D \end{bmatrix}.$$

[078] In many instances, the formulas and algorithm shown above identify the same model (same values for A, B, C, and D) regardless of the R_2 area 608 selected. In other instances, such as when a monitored system 102 suffers from drift, a validation step may be used to remove this undesired effect on the quality of the model selected. For example, in particular embodiments, the following equation is used during the validation step:

$$\min_{p_i \in \{p_1, \dots, p_n\}} \|R_{E3}\|_2^2$$

where p_i represents the i -th pole of the pole candidate set, and R_{E3} is a function of the model parameters A, B, C, and D. As shown in FIGURE 6C, R_{E3} can be identified by performing CQR decomposition on a matrix formed by a backward Hankel matrix (U_b) of the input signal 104 and a forward Hankel matrix ($Y - \hat{Y}$) that is made of the prediction error ($y - \hat{y}$).

[079] Although FIGURES 6A through 6C illustrate examples of canonical QR-decompositions used for model identification, various changes may be made to FIGURES 6A through 6C. For example, any number of R_2 areas 608 and U_{fb} matrices 610 may be used. Also, the validation step described above may be performed

during every model identification, during none of the model identifications, or during some of the model identifications. In addition, other techniques for using the projections of FIGURES 4A through 4E to model relationships between signals could be used.

[080] FIGURE 7 illustrates an example method 700 for modeling relationships between signals according to one embodiment of this disclosure. For ease of explanation, the method 700 is described with respect to the controller 112 operating in the system 100 of FIGURE 1. The method 700 could be used by any other apparatus or device in any system.

[081] The controller 112 forms a projection associated with two or more signals at step 702. This may include, for example, the controller 112 generating a projection as shown in one of FIGURES 4A through 4E using the techniques described above.

[082] The controller 112 selects one or more regions in the projection at step 704. This may include, for example, the controller 112 identifying one or more areas 608 in the projection. The controller 112 could select one or multiple areas 608 based, for example, on user input, a default number and definition of the areas 608, or in any other suitable manner.

[083] The controller 112 identifies one or more pole candidates for the model using the projection at step 706. This

may include, for example, the controller 112 using the algorithm shown above in Paragraph [074] to identify possible values for the poles. This may also include the controller 112 using the selected area(s) of the projection to identify the possible poles. The controller 112 could use any other suitable technique to identify values for the poles.

[084] The controller 112 identifies one or more model candidates for the model using the projection at step 708. This may include, for example, the controller 112 using the formulas shown above in Paragraphs [076] and [077] to identify values for the model candidates. This may also include the controller 112 using the selected area(s) of the projection and various information generated during identification of the pole candidates to identify the model candidates. The controller 112 could use any other suitable technique to identify values for the model candidates.

[085] The controller 112 performs model validation and order reduction if necessary at step 710. This may include, for example, the controller 112 using the validation step described above in Paragraph [078] to validate the identified model. This may also include the controller 112 performing system-order reduction to reduce the order of the identified model. However, as described above, the same model may be produced regardless of

which R_2 area 608 is used by the controller 112 in particular situations. As a result, in these situations, the controller 112 could skip step 710.

[086] At this point, the controller 112 could use the identified model in any suitable manner. For example, the controller 112 could use the model to "de-noise" the actual output signal 110, which is labeled Y . As a particular example, the controller 112 could identify a model having the highest order as plausible. The controller 112 then uses the model to predict what the actual output signal 110 would look like without any noise or other disturbances 108. The predicted signal is referred to as \hat{Y} . The controller 112 then defines the noise or drift called e in the actual output signal 110 using the formula:

$$e = Y - \hat{Y}, \text{ or}$$

$$\hat{Y} = Y - E.$$

Here, the signal defined by \hat{Y} can be explained by the input signal 104, and the noise or drift defined by e is not explained by the input signal 104. This represents one example use of the identified model. The controller 112 could use the identified model in any other suitable manner.

[087] Although FIGURE 7 illustrates one example of a method 700 for modeling relationships between signals, various changes may be made to FIGURE 7. For example, the controller 112 could

receive a projection produced by another component in the system 100 and process the projection. In these embodiments, the controller 112 need not generate the projection at step 702.

[088] It may be advantageous to set forth definitions of certain words and phrases used throughout this patent document. The terms "include" and "comprise," as well as derivatives thereof, mean inclusion without limitation. The term "or" is inclusive, meaning and/or. The phrases "associated with" and "associated therewith," as well as derivatives thereof, may mean to include, be included within, interconnect with, contain, be contained within, connect to or with, couple to or with, be communicable with, cooperate with, interleave, juxtapose, be proximate to, be bound to or with, have, have a property of, or the like. The term "controller" means any device, system or part thereof that controls at least one operation. A controller may be implemented in hardware, firmware, software, or some combination of at least two of the same. The functionality associated with any particular controller may be centralized or distributed, whether locally or remotely.

[089] While this disclosure has described certain embodiments and generally associated methods, alterations and permutations of these embodiments and methods will be apparent to those skilled in the art. Accordingly, the above description of example

embodiments does not define or constrain this disclosure. Other changes, substitutions, and alterations are also possible without departing from the spirit and scope of this disclosure, as defined by the following claims.

SOFTWARE APPENDIX

```

function [Q,R]=CQR_H(A,save)

% Usage: [Q,R]=CQR_H(A)
%           Q=CQR_H(A)
% This is CQR Household algorithm. It is as economical as
% the standard QR Household algorithm.

%for testing the algorithm accuracy
A0=A;

[n,m]=size(A);
n1=n+1;
%% trianglize A
mm=min(m,n-1);
for j=1:mm
    v=HouseHld(A(j:n,j));
    A(j:n,j:m)=HousePre(A(j:n,j:m),v);
    A(j+1:n,j)=v(2:(n1-j));
end

if nargout <= 1
    Q=A;
else
    if nargin < 2 | n <= m
        R=zeros(size(A));
        Q=eye(n);
        ncol = n;
    elseif save == 0 & n > m
        R=zeros(m,m);
        Q=[eye(m);zeros(n-m,m)];
        ncol = m;
    else
        error('input format error'),
    end

    for j=mm:-1:1
        v=[1;A(j+1:n,j)];
        Q(j:n,j:ncol)=HousePre(Q(j:n,j:ncol),v);
        R(1:j,j)=A(1:j,j);

        if R(j,j) < 0;
            R(j,j:m) = -R(j,j:m);
            if nargout > 1
                Q(j:n,j) = -Q(j:n,j);
            end
        end
    end
end

```

```
        end
      end
    end

    for j=mm+1:m
      R(1:n,j)=A(1:n,j);
    end

    if m >= n & R(n, n) < 0;
      R(n,n:m) = -R(n,n:m);
      if nargout > 1
        Q(:,n) = -Q(:,n);
      end
    end
  end
end
```

```
function [v, P]=HouseHld(x,i)
```

```
%      v=HouseHld(x,i)
```

```
n=length(x);
```

```
nx=norm(x);
```

```
v=zeros(size(x));
```

```
if nargin == 1, i=1; end
```

```
ind=[1:i-1,i+1:n];
```

```
if nx > eps
```

```
    b=x(i)+sign(x(i))*nx;
```

```
    v(ind)=x(ind)/b;
```

```
else
```

```
    v(ind)=x(ind);
```

```
end
```

```
v(i)=1;
```

```
if nargin > 1
```

```
    P=eye(n) - (2*v)*(v'/(v'*v));
```

```
end
```

```
function A=HousePre(A,v)
```

```
% Usage: Ap=HousePre(A,v)
```

```
% Pre-multiply the Householder transformation P(v) to A
```

```
%      Ap = P(v)*A
```

```
A = A + ((-2/(v'*v))*v)*(v'*A);
```

```
%A = A - ((2/(v'*v))*v)*(v'*A);
```

```
% = (I -2/(v'*v)*(v*v')) * A
```

```
% thus,
```

```
% P(v) = I -2/(v'*v)*(v*v')  -> symmetric
```